Defining and using connector in ModelingToolkit Julia

1. Load packages

In [2]:
```
using Pkg
Pkg.add(["ModelingToolkit", "OrdinaryDiffEq", "DiffEqCallbacks", "Plots"]);
```

```
   Resolving package versions...
     Project No packages added to or removed from `C:\Users\Firstname Lastname\.juli
a\environments\v1.12\Project.toml`
    Manifest No packages added to or removed from `C:\Users\Firstname Lastname\.juli
a\environments\v1.12\Manifest.toml`
```

In [3]:
```
using ModelingToolkit, OrdinaryDiffEq, DiffEqCallbacks, Plots # req packages
using ModelingToolkit: t_nounits as t, D_nounits as D;
```

2. Connector

> Suppose we want to define a **power connector** or **energy exchange point**, denoted as `Con`, for a physical domain such as electrical, mechanical, or hydraulic.
>
> This connector carries two fundamental **power variables**: an **effort variable** (e(t)) and a **flow variable** (f(t)).
>
> Conceptually, the connector `Con` represents a **partial model** — a *base building block* used to model multiphysics systems. Different subsystems interact and exchange energy through this connector by satisfying conservation laws.
>
> In **ModelingToolkit.jl**, the connector `Con` can be defined as:

In [4]:
```
@connector Con begin
    e(t)
    f(t), [connect = Flow]
end
```

Out[4]:
```
ModelingToolkit.Model{typeof(__Con__), Dict{Symbol, Any}}(Main.__Con__, Dict{Symbo
l, Any}(:metadata => Dict{Symbol, Any}(), :variables => Dict{Symbol, Dict{Symbol,
Any}}(:f => Dict(:type => Real, :connection_type => :Flow), :e => Dict(:type => Re
al)), :kwargs => Dict{Symbol, Dict}(:f => Dict{Symbol, Union{Nothing, DataType}}(:
value => nothing, :type => Real), :e => Dict{Symbol, Union{Nothing, DataType}}(:va
lue => nothing, :type => Real)), :independent_variable => :t), true)
```

Connectotr can be defined using @named macro

In [5]:
```
@named con = Con()
```

Out[5]:

Let us see how many equations and unknowns the con model has using @show macro

---

In [6]:
```
@show con
```

```
con = Model con:
Equations (2):
  2 connecting: see equations(expand_connections(con))
Unknowns (2): see unknowns(con)
  e(t)
  f(t)
```

Out[6]:

So we see two equations and two unknowns. Equations can be viewed as equations(con) and expand_connections(con) will provide hidden equations. unknowns(con) gives all the variables used inside the model.

In [7]:
```
#
equations(expand_connections(con))
```

Out[7]:

Since connector is a partial model of the ODE ecosystem in ModelingToolkit that contains variables and not equations, it is obvious to not see equations of the connector con. Later when equations are added by extending these connector models we will see the equations that build component such as Resistor and Capacitor in an electrical system modeling work.

In [8]:
```
#
unknowns(con)
```

Out[8]:
```
2-element Vector{SymbolicUtils.BasicSymbolic{Real}}:
 e(t)
 f(t)
```

We can also inspect the effort and flow variable of the connector using the dot operator as below:

In [9]:
```
#
con.e
```

Out[9]:

$$\texttt{con.e}\left(t\right) \tag{1}$$

In [10]:
```
#
con.f
```

Out[10]:

$$\texttt{con.f}\,(t) \tag{2}$$

Note: A connector only has a effort and a flow variable. A connector in a equation based modeling lanaguage is a partial model formed using effort and flow variables, which are easily inherited into object oriented programming concept to later complete the whole model combining these kinds of other partial model.

--- break ---

Let us connect two connectors

In [11]:
```
@named begin
    con1 = Con()
    con2 = Con()
end
```

Out[11]:
```
2-element Vector{System}:
 Model con1:
Equations (2):
  2 connecting: see equations(expand_connections(con1))
Unknowns (2): see unknowns(con1)
  e(t)
  f(t)
 Model con2:
Equations (2):
  2 connecting: see equations(expand_connections(con2))
Unknowns (2): see unknowns(con2)
  e(t)
  f(t)
```

In [12]:
```
equations(expand_connections(con1))
```

Out[12]:

In [13]:
```
eqs = connect(con1, con2)
```

Out[13]:

$$\text{connect}\,(con1, con2) \tag{3}$$

In [14]:
```
@show eqs
```

```
eqs = connect(con1, con2)
```

Out[14]:

$$\text{connect}\left(con1, con2\right) \tag{4}$$

We see that by connecting two connector we do not have any equations. It is an empty equation ODE system for the ModelingToolkit to process.

In [15]:
```
@mtkcompile con1 = Con()
```

Out[15]:

---

3. Defining a connector for a single port system

A single-port system (or 1-port) has one energy flow path — one input and one output.
Examples:Electrical resistor, inductor, capacitor, Hydraulic orifice, Mechanical damper, etc.

In [16]:
```
@mtkmodel OnePort begin
    @components {pCon = Con(); nCon = Con()}
    @variables {e(t); f(t)}
    @equations {e ~ pCon.e - nCon.e; 0 ~ pCon.f + nCon.f; f ~ pCon.f}
end
;
```

Let us inspect OnePort partial model using @named and @show macros also let us see total equations both standard and connecting equations.

In [17]:
```
@named op = OnePort()
```

Out[17]:

$$e\left(t\right) = -\mathtt{nCon.e}\left(t\right) + \mathtt{pCon.e}\left(t\right) \tag{5}$$
$$0 = \mathtt{pCon.f}\left(t\right) + \mathtt{nCon.f}\left(t\right) \tag{6}$$
$$f\left(t\right) = \mathtt{pCon.f}\left(t\right) \tag{7}$$

In [18]:
```
@show op
```

```
op = Model op:
Subsystems (2): see hierarchy(op)
  pCon
  nCon
Equations (5):
  3 standard: see equations(op)
  2 connecting: see equations(expand_connections(op))
Unknowns (6): see unknowns(op)
  e(t)
  f(t)
  pCon₊e(t)
  pCon₊f(t)
  nCon₊e(t)
  nCon₊f(t)
```

Out[18]:

$$e(t) = -\mathtt{nCon.e}(t) + \mathtt{pCon.e}(t) \tag{8}$$
$$0 = \mathtt{pCon.f}(t) + \mathtt{nCon.f}(t) \tag{9}$$
$$f(t) = \mathtt{pCon.f}(t) \tag{10}$$

In [19]:
```
expand_connections(op)
```

Out[19]:

$$e(t) = -\mathtt{nCon.e}(t) + \mathtt{pCon.e}(t) \tag{11}$$
$$0 = \mathtt{pCon.f}(t) + \mathtt{nCon.f}(t) \tag{12}$$
$$f(t) = \mathtt{pCon.f}(t) \tag{13}$$
$$0 = \mathtt{pCon.f}(t) \tag{14}$$
$$0 = \mathtt{nCon.f}(t) \tag{15}$$

In [20]:
```
unknowns(op)
```

Out[20]:
```
6-element Vector{SymbolicUtils.BasicSymbolic{Real}}:
 e(t)
 f(t)
 pCon₊e(t)
 pCon₊f(t)
 nCon₊e(t)
 nCon₊f(t)
```

We see that the OnePort partial model has 5 equations but 6 variables. pCon and nCon are positive and negative connectors, respectively, inherited from the connector Con. Partial OnePort model will be used while defining model of a device in a system such as resistor or capacitor.

---

4. Defining a connector for a two port system

A two-port system (or 2-port) has two energy flow path — two input and two output such as a Transformer.

In [22]:
```
@mtkmodel Transformer begin
    @components begin
        op1 = OnePort()
        op2 = OnePort()
    end
    @parameters begin
        k = 0.9     # coupling coefficient
        N1 = 100
        N2 = 200
    end
    @equations begin
        # Ideal transformer relation
        op2.e ~ (N2/N1) * op1.e
        op2.f ~ -(N1/N2) * op1.f
    end
end
```

Out[22]:
```
ModelingToolkit.Model{typeof(__Transformer__), Dict{Symbol, Any}}(Main.__Transform
er__, Dict{Symbol, Any}(:metadata => Dict{Symbol, Any}(), :components => Any[Union
{Expr, Symbol}[:op1, :OnePort], Union{Expr, Symbol}[:op2, :OnePort]], :kwargs => D
ict{Symbol, Dict}(:k => Dict{Symbol, Any}(:value => 0.9, :type => Real), :N2 => Di
ct{Symbol, Any}(:value => 200, :type => Real), :N1 => Dict{Symbol, Any}(:value =>
100, :type => Real)), :independent_variable => t, :parameters => Dict{Symbol, Dict
{Symbol, Any}}(:k => Dict(:default => 0.9, :type => Real), :N2 => Dict(:default =>
200, :type => Real), :N1 => Dict(:default => 100, :type => Real)), :equations => A
ny["op2.e ~ (N2 / N1) * op1.e", "op2.e ~ (N2 / N1) * op1.e", "op2.f ~ -(N1 / N2) *
op1.f"]), false)
```

In [23]:
```
@named TX = Transformer()
@show TX
```

```
TX = Model TX:
Subsystems (2): see hierarchy(TX)
  op1
  op2
Equations (12):
  8 standard: see equations(TX)
  4 connecting: see equations(expand_connections(TX))
Unknowns (12): see unknowns(TX)
  op1₊e(t)
  op1₊f(t)
  op1₊pCon₊e(t)
  op1₊pCon₊f(t)
  op1₊nCon₊e(t)
  op1₊nCon₊f(t)
  op2₊e(t)
  op2₊f(t)
  op2₊pCon₊e(t)
  op2₊pCon₊f(t)
  op2₊nCon₊e(t)
  op2₊nCon₊f(t)
Parameters (3): see parameters(TX)
  k [defaults to 0.9]
  N1 [defaults to 100]
  N2 [defaults to 200]
```

Out[23]:

$$\mathtt{op2.e}\left(t\right) = \frac{\mathtt{N2op1.e}\left(t\right)}{\mathtt{N1}} \tag{16}$$
$$\mathtt{op2.f}\left(t\right) = -\mathtt{op1.f}\left(t\right)\frac{\mathtt{N1}}{\mathtt{N2}} \tag{17}$$
$$\mathtt{op1.e}\left(t\right) = -\mathtt{op1.nCon.e}\left(t\right) + \mathtt{op1.pCon.e}\left(t\right) \tag{18}$$
$$0 = \mathtt{op1.nCon.f}\left(t\right) + \mathtt{op1.pCon.f}\left(t\right) \tag{19}$$
$$\mathtt{op1.f}\left(t\right) = \mathtt{op1.pCon.f}\left(t\right) \tag{20}$$
$$\mathtt{op2.e}\left(t\right) = \mathtt{op2.pCon.e}\left(t\right) - \mathtt{op2.nCon.e}\left(t\right) \tag{21}$$
$$0 = \mathtt{op2.pCon.f}\left(t\right) + \mathtt{op2.nCon.f}\left(t\right) \tag{22}$$
$$\mathtt{op2.f}\left(t\right) = \mathtt{op2.pCon.f}\left(t\right) \tag{23}$$

5. Electrical Circuit System

Let us consider that we want to model circuits having R, L, and C components. It is also important to develop model for ground and voltage source.

In electrical system, voltage is the effort variable and current is the flow variable.

Let us define a connector called a **Pin** that is a point in the network where energy is exchange when connecting several components together.

In [24]:
```
@connector Pin begin
    v(t)
    i(t), [connect = Flow]
end
;
```

The electrical subsystem Ground has zero potential and can be modeled by extending a Pin model as below:

In [25]:
```
@mtkmodel Ground begin
    @components {g = Pin()}
    @equations {g.v ~ 0}
end
;
```

In [26]:
```
@named gnd = Ground()
@show gnd
```

```
gnd = Model gnd:
Subsystems (1): see hierarchy(gnd)
  g
Equations (2):
  1 standard: see equations(gnd)
  1 connecting: see equations(expand_connections(gnd))
Unknowns (2): see unknowns(gnd)
  g₊v(t)
  g₊i(t)
```

Out[26]:

$$\mathrm{g.v}(t) = 0 \tag{24}$$

OnePort can be definded inheritance of the two pins.

In [27]:
```
@mtkmodel OnePort begin
    @components {p = Pin(); n = Pin()}
    @variables {v(t); i(t)}
    @equations {v ~ p.v - n.v; 0 ~ p.i + n.i; i ~ p.i}
end
;
```

In [28]:
```
@named op = OnePort()
```

Out[28]:

$$v(t) = \mathrm{p.v}(t) - \mathrm{n.v}(t) \tag{25}$$
$$0 = \mathrm{n.i}(t) + \mathrm{p.i}(t) \tag{26}$$
$$i(t) = \mathrm{p.i}(t) \tag{27}$$

In [29]:
```
@show op
```

```
op = Model op:
Subsystems (2): see hierarchy(op)
  p
  n
Equations (5):
  3 standard: see equations(op)
  2 connecting: see equations(expand_connections(op))
Unknowns (6): see unknowns(op)
  v(t)
  i(t)
  p₊v(t)
  p₊i(t)
  n₊v(t)
  n₊i(t)
```

Out[29]:

$$v(t) = \mathrm{p.v}(t) - \mathrm{n.v}(t) \tag{28}$$
$$0 = \mathrm{n.i}(t) + \mathrm{p.i}(t) \tag{29}$$
$$i(t) = \mathrm{p.i}(t) \tag{30}$$

In [31]:
```
equations(expand_connections(op))
```

Out[31]:

$$v(t) = \mathtt{p.v}(t) - \mathtt{n.v}(t) \tag{31}$$
$$0 = \mathtt{n.i}(t) + \mathtt{p.i}(t) \tag{32}$$
$$i(t) = \mathtt{p.i}(t) \tag{33}$$
$$0 = \mathtt{p.i}(t) \tag{34}$$
$$0 = \mathtt{n.i}(t) \tag{35}$$

Some Observations

1. Connector has zero equations and defined as a energy exchange point.
2. OnePort which is extended from connector using two connector points has 5 equations and 6 unknowns
3. OnePort will be extended to form a device such as resistor, capacitor, etc. The devices will have equal number of unknowns and equations.

Device Modeling

The modelingtoolkit model **Pin** has only two varibles. The model **OnePort** has six variables and 5 equations. Next, we wil use this **OnePort** to form a device model i.e. system component level model such as resistor, capacitor and inductor.

Resistor

In [32]:
```
@mtkmodel Resistor begin
    @extend OnePort()
    @parameters {R = 1.0}
    @equations {v ~ i*R}
end
;
```

In [33]:
```
@named rg = Resistor(R=2.0)
@show rg
```

```
rg = Model rg:
Subsystems (2): see hierarchy(rg)
  p
  n
Equations (6):
  4 standard: see equations(rg)
  2 connecting: see equations(expand_connections(rg))
Unknowns (6): see unknowns(rg)
  v(t)
  i(t)
  p₊v(t)
  p₊i(t)
  n₊v(t)
  n₊i(t)
Parameters (1): see parameters(rg)
  R [defaults to 2.0]
```

Out[33]:

$$v(t) = \mathtt{p.v}(t) - \mathtt{n.v}(t) \tag{36}$$
$$0 = \mathtt{n.i}(t) + \mathtt{p.i}(t) \tag{37}$$
$$i(t) = \mathtt{p.i}(t) \tag{38}$$
$$v(t) = Ri(t) \tag{39}$$

In [33]:
```
equations(expand_connections(rg))
```

Out[33]:

$$v(t) = \mathtt{p.v}(t) - \mathtt{n.v}(t) \tag{40}$$
$$0 = \mathtt{n.i}(t) + \mathtt{p.i}(t) \tag{41}$$
$$i(t) = \mathtt{p.i}(t) \tag{42}$$
$$v(t) = Ri(t) \tag{43}$$
$$0 = \mathtt{p.i}(t) \tag{44}$$
$$0 = \mathtt{n.i}(t) \tag{45}$$

Inductor

In [36]:
```
@mtkmodel Inductor begin
    @extend OnePort()
    @parameters {L = 1.0}
    @equations {D(i) ~ v/L}
end
;

@named ind = Inductor()
@show ind
equations(expand_connections(ind))
```

```
ind = Model ind:
Subsystems (2): see hierarchy(ind)
  p
  n
Equations (6):
  4 standard: see equations(ind)
  2 connecting: see equations(expand_connections(ind))
Unknowns (6): see unknowns(ind)
  v(t)
  i(t)
  p₊v(t)
  p₊i(t)
  n₊v(t)
  n₊i(t)
Parameters (1): see parameters(ind)
  L [defaults to 1.0]
```

Out[36]:

$$v(t) = \mathtt{p.v}(t) - \mathtt{n.v}(t) \tag{46}$$
$$0 = \mathtt{n.i}(t) + \mathtt{p.i}(t) \tag{47}$$
$$i(t) = \mathtt{p.i}(t) \tag{48}$$
$$\frac{\mathrm{d}i(t)}{\mathrm{d}t} = \frac{v(t)}{L} \tag{49}$$
$$0 = \mathtt{p.i}(t) \tag{50}$$
$$0 = \mathtt{n.i}(t) \tag{51}$$

Capacitor

In [35]:

```
@mtkmodel Capacitor begin
    @extend OnePort()
    @parameters {C = 1.0}
    @equations {D(v) ~ i/C}
end
;
@named cap = Capacitor()
@show cap
equations(expand_connections(cap))
```

```
cap = Model cap:
Subsystems (2): see hierarchy(cap)
  p
  n
Equations (6):
  4 standard: see equations(cap)
  2 connecting: see equations(expand_connections(cap))
Unknowns (6): see unknowns(cap)
  v(t)
  i(t)
  p₊v(t)
  p₊i(t)
  n₊v(t)
  n₊i(t)
Parameters (1): see parameters(cap)
  C [defaults to 1.0]
```

Out[35]:

$$v(t) = \mathtt{p.v}(t) - \mathtt{n.v}(t) \tag{52}$$
$$0 = \mathtt{n.i}(t) + \mathtt{p.i}(t) \tag{53}$$
$$i(t) = \mathtt{p.i}(t) \tag{54}$$
$$\frac{\mathrm{d}v(t)}{\mathrm{d}t} = \frac{i(t)}{C} \tag{55}$$
$$0 = \mathtt{p.i}(t) \tag{56}$$
$$0 = \mathtt{n.i}(t) \tag{57}$$

Constant voltage source

In [37]:
```
@mtkmodel ConstantVoltage begin
    @extend OnePort()
    @parameters {V = 1.0}
    @equations {v ~ V} # enforce source voltage across its pins
end
;
@named vol = ConstantVoltage()
@show vol
equations(expand_connections(vol))
```

```
vol = Model vol:
Subsystems (2): see hierarchy(vol)
  p
  n
Equations (6):
  4 standard: see equations(vol)
  2 connecting: see equations(expand_connections(vol))
Unknowns (6): see unknowns(vol)
  v(t)
  i(t)
  p₊v(t)
  p₊i(t)
  n₊v(t)
  n₊i(t)
Parameters (1): see parameters(vol)
  V [defaults to 1.0]
```

Out[37]:

$$
\begin{aligned}
v\left(t\right) &= \mathtt{p.v}\left(t\right) - \mathtt{n.v}\left(t\right) && (58)\\
0 &= \mathtt{n.i}\left(t\right) + \mathtt{p.i}\left(t\right) && (59)\\
i\left(t\right) &= \mathtt{p.i}\left(t\right) && (60)\\
v\left(t\right) &= V && (61)\\
0 &= \mathtt{p.i}\left(t\right) && (62)\\
0 &= \mathtt{n.i}\left(t\right) && (63)
\end{aligned}
$$

6. Modeling RLC type circuits

R Model

In [38]:
```
@mtkmodel RModel begin
    @description "A resistive circuit."
    @components {vol = ConstantVoltage(V = 1.0); res = Resistor(R=1.0); gnd = Groun
    @equations {connect(vol.p,res.p); connect(res.n,vol.n); connect(res.n,gnd.g)}
end
;
```

In [39]:
```
@named r_model = RModel()
```

Out[39]:

$$
\begin{bmatrix}
\text{connect}\left(vol_{+}p, res_{+}p\right)\\
\text{connect}\left(res_{+}n, vol_{+}n\right)\\
\text{connect}\left(res_{+}n, gnd_{+}g\right)\\
\mathtt{vol.v}\left(t\right) = \mathtt{vol.p.v}\left(t\right) - \mathtt{vol.n.v}\left(t\right)\\
0 = \mathtt{vol.n.i}\left(t\right) + \mathtt{vol.p.i}\left(t\right)\\
\mathtt{vol.i}\left(t\right) = \mathtt{vol.p.i}\left(t\right)\\
\mathtt{vol.v}\left(t\right) = \mathtt{vol.V}\\
\mathtt{res.v}\left(t\right) = -\mathtt{res.n.v}\left(t\right) + \mathtt{res.p.v}\left(t\right)\\
0 = \mathtt{res.n.i}\left(t\right) + \mathtt{res.p.i}\left(t\right)\\
\mathtt{res.i}\left(t\right) = \mathtt{res.p.i}\left(t\right)\\
\mathtt{res.v}\left(t\right) = \mathtt{res.R}\,\mathtt{res.i}\left(t\right)\\
\mathtt{gnd.g.v}\left(t\right) = 0
\end{bmatrix}
\tag{64}
$$

In [40]:
```
@show r_model
```

```
r_model = Model r_model: A resistive circuit.
Subsystems (3): see hierarchy(r_model)
  vol
  res
  gnd
Equations (14):
  9 standard: see equations(r_model)
  5 connecting: see equations(expand_connections(r_model))
Unknowns (14): see unknowns(r_model)
  vol₊v(t)
  vol₊i(t)
  vol₊p₊v(t)
  vol₊p₊i(t)
  vol₊n₊v(t)
  vol₊n₊i(t)
  res₊v(t)
  res₊i(t)
  res₊p₊v(t)
  res₊p₊i(t)
  res₊n₊v(t)
  res₊n₊i(t)
  gnd₊g₊v(t)
  gnd₊g₊i(t)
Parameters (2): see parameters(r_model)
  vol₊V [defaults to 1.0]
  res₊R [defaults to 1.0]
```

Out[40]:

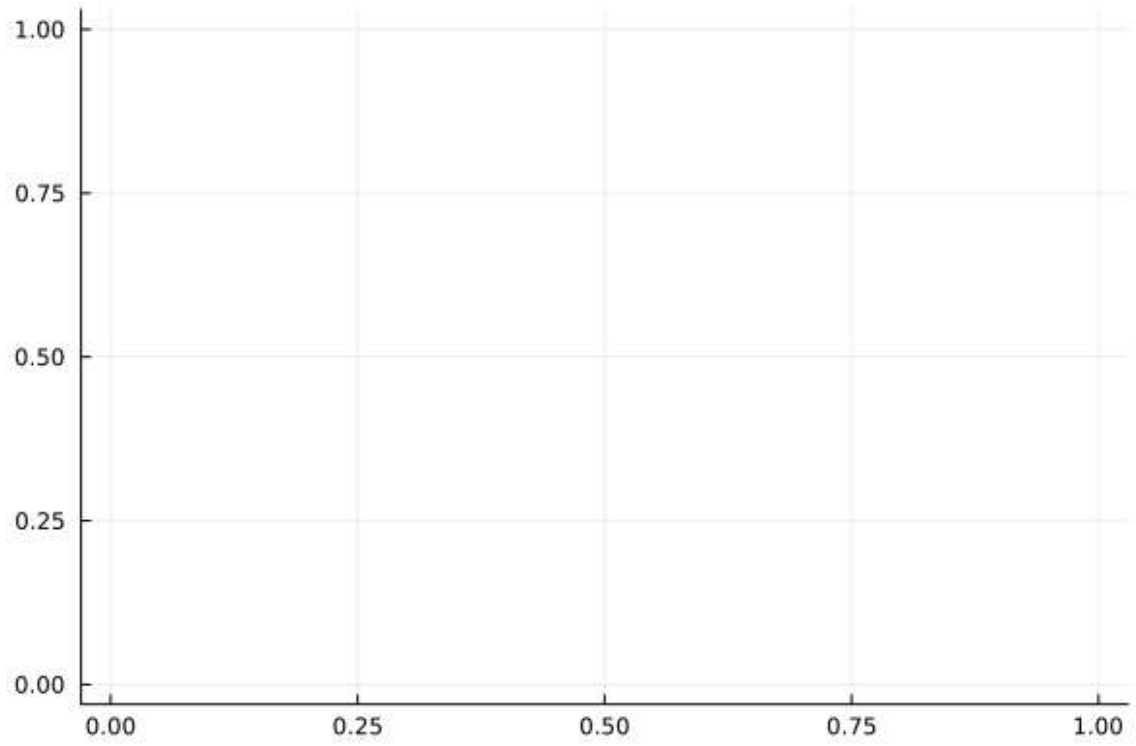
$$\begin{bmatrix} \text{connect}\left(vol_{+}p, res_{+}p\right) \\ \text{connect}\left(res_{+}n, vol_{+}n\right) \\ \text{connect}\left(res_{+}n, gnd_{+}g\right) \\ \mathtt{vol.v}\left(t\right) = \mathtt{vol.p.v}\left(t\right) - \mathtt{vol.n.v}\left(t\right) \\ 0 = \mathtt{vol.n.i}\left(t\right) + \mathtt{vol.p.i}\left(t\right) \\ \mathtt{vol.i}\left(t\right) = \mathtt{vol.p.i}\left(t\right) \\ \mathtt{vol.v}\left(t\right) = \mathtt{vol.V} \\ \mathtt{res.v}\left(t\right) = -\mathtt{res.n.v}\left(t\right) + \mathtt{res.p.v}\left(t\right) \\ 0 = \mathtt{res.n.i}\left(t\right) + \mathtt{res.p.i}\left(t\right) \\ \mathtt{res.i}\left(t\right) = \mathtt{res.p.i}\left(t\right) \\ \mathtt{res.v}\left(t\right) = \mathtt{res.Rres.i}\left(t\right) \\ \mathtt{gnd.g.v}\left(t\right) = 0 \end{bmatrix} \tag{65}$$

In [41]:
```
@mtkcompile r_model = RModel()
```

Out[41]:

In [42]:
```
u0 = []
prob = ODEProblem(r_model, u0, (0, 10.0))
sol = solve(prob)
plot(sol)
```

Out[42]:



In [43]:
```
sol
```

Out[43]:
```
retcode: Success
Interpolation: 1st order linear
t: 2-element Vector{Float64}:
  0.0
 10.0
u: 2-element Vector{Vector{Float64}}:
 []
 []
```

Series RL Model

In [46]:
```
@mtkmodel RLModel begin
    @description "A resistive circuit."
    @components {vol = ConstantVoltage(V = 1.0); res = Resistor(R=1.0);ind = Induct
    @equations {connect(vol.p,res.p); connect(res.n,ind.p); connect(ind.n,vol.n); c
end
;
```

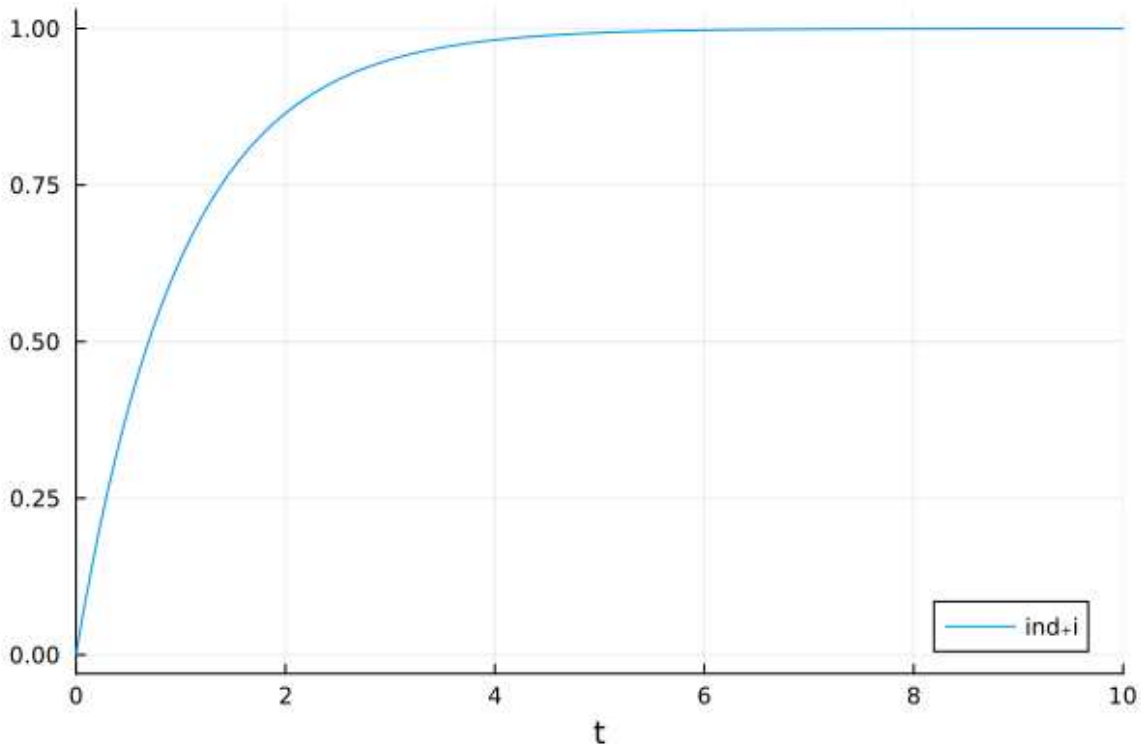
In [47]:
```
@mtkcompile rl_model = RLModel()
```

Out[47]:

$$\frac{\mathrm{d}\mathtt{ind.i}(t)}{\mathrm{d}t} = \frac{\mathtt{ind.v}(t)}{\mathtt{ind.L}} \tag{66}$$

In [50]:
```
u0 = [rl_model.ind.i => 0.0]
prob = ODEProblem(rl_model, u0, (0, 10.0))
sol = solve(prob)
plot(sol)
```

Out[50]:



Series RLC

In [52]:
```
@mtkmodel RLCModel begin
    @description "A circuit with a constant voltage source, resistor and capacitor
    @components begin
        resistor = Resistor(R = 1.0)
        inductor = Inductor(L = 1.0)
        capacitor = Capacitor(C = 1.0)
        source = ConstantVoltage(V = 1.0)
        ground = Ground()
    end
    @equations begin
        connect(source.p, resistor.p)
        connect(resistor.n, inductor.p)
        connect(inductor.n, capacitor.p)
        connect(capacitor.n, source.n)
        connect(source.n, ground.g)
    end
end
;
```

In [54]:
```
@named rlc_model = RLCModel()
@show rlc_model
```

```
rlc_model = Model rlc_model: A circuit with a constant voltage source, resistor and
capacitor connected in series.
Subsystems (5): see hierarchy(rlc_model)
  resistor
  inductor
  capacitor
  source
  ground
Equations (26):
  17 standard: see equations(rlc_model)
  9 connecting: see equations(expand_connections(rlc_model))
Unknowns (26): see unknowns(rlc_model)
  resistor₊v(t)
  resistor₊i(t)
  resistor₊p₊v(t)
  resistor₊p₊i(t)
  resistor₊n₊v(t)
  resistor₊n₊i(t)
  inductor₊v(t)
  inductor₊i(t)
  inductor₊p₊v(t)
  inductor₊p₊i(t)
  inductor₊n₊v(t)
  inductor₊n₊i(t)
  capacitor₊v(t)
  capacitor₊i(t)
  capacitor₊p₊v(t)
  capacitor₊p₊i(t)
  capacitor₊n₊v(t)
  capacitor₊n₊i(t)
  source₊v(t)
  source₊i(t)
  source₊p₊v(t)
  source₊p₊i(t)
  source₊n₊v(t)
  source₊n₊i(t)
  ground₊g₊v(t)
  ground₊g₊i(t)
Parameters (4): see parameters(rlc_model)
  resistor₊R [defaults to 1.0]
  inductor₊L [defaults to 1.0]
  capacitor₊C [defaults to 1.0]
  source₊V [defaults to 1.0]
```

Out[54]:

$$\begin{bmatrix} \text{connect}\left(source_{+}p, resistor_{+}p\right) \\ \text{connect}\left(resistor_{+}n, inductor_{+}p\right) \\ \text{connect}\left(inductor_{+}n, capacitor_{+}p\right) \\ \text{connect}\left(capacitor_{+}n, source_{+}n\right) \\ \text{connect}\left(source_{+}n, ground_{+}g\right) \\ \texttt{resistor.v}\left(t\right) = -\texttt{resistor.n.v}\left(t\right) + \texttt{resistor.p.v}\left(t\right) \\ 0 = \texttt{resistor.p.i}\left(t\right) + \texttt{resistor.n.i}\left(t\right) \\ \texttt{resistor.i}\left(t\right) = \texttt{resistor.p.i}\left(t\right) \\ \texttt{resistor.v}\left(t\right) = \texttt{resistor.Rresistor.i}\left(t\right) \\ \texttt{inductor.v}\left(t\right) = -\texttt{inductor.n.v}\left(t\right) + \texttt{inductor.p.v}\left(t\right) \\ 0 = \texttt{inductor.p.i}\left(t\right) + \texttt{inductor.n.i}\left(t\right) \\ \texttt{inductor.i}\left(t\right) = \texttt{inductor.p.i}\left(t\right) \\ \frac{\mathrm{d}\texttt{inductor.i}(t)}{\mathrm{d}t} = \frac{\texttt{inductor.v}(t)}{\texttt{inductor.L}} \\ \texttt{capacitor.v}\left(t\right) = \texttt{capacitor.p.v}\left(t\right) - \texttt{capacitor.n.v}\left(t\right) \\ 0 = \texttt{capacitor.p.i}\left(t\right) + \texttt{capacitor.n.i}\left(t\right) \\ \texttt{capacitor.i}\left(t\right) = \texttt{capacitor.p.i}\left(t\right) \\ \frac{\mathrm{d}\texttt{capacitor.v}(t)}{\mathrm{d}t} = \frac{\texttt{capacitor.i}(t)}{\texttt{capacitor.C}} \\ \texttt{source.v}\left(t\right) = -\texttt{source.n.v}\left(t\right) + \texttt{source.p.v}\left(t\right) \\ 0 = \texttt{source.p.i}\left(t\right) + \texttt{source.n.i}\left(t\right) \\ \texttt{source.i}\left(t\right) = \texttt{source.p.i}\left(t\right) \\ \texttt{source.v}\left(t\right) = \texttt{source.V} \\ \texttt{ground.g.v}\left(t\right) = 0 \end{bmatrix} \tag{67}$$

In [55]:
```
@mtkcompile rlc_model = RLCModel()
```

Out[55]:

$$\frac{\mathrm{d}\texttt{capacitor.v}\left(t\right)}{\mathrm{d}t} = \frac{\texttt{capacitor.i}\left(t\right)}{\texttt{capacitor.C}} \tag{68}$$

$$\frac{\mathrm{d}\texttt{inductor.i}\left(t\right)}{\mathrm{d}t} = \frac{\texttt{inductor.v}\left(t\right)}{\texttt{inductor.L}} \tag{69}$$

In [56]:
```
@show rlc_model
```

```
rlc_model = Model rlc_model: A circuit with a constant voltage source, resistor and
capacitor connected in series.
Equations (2):
  2 standard: see equations(rlc_model)
Unknowns (2): see unknowns(rlc_model)
  capacitor₊v(t)
  inductor₊i(t)
Parameters (4): see parameters(rlc_model)
  resistor₊R [defaults to 1.0]
  inductor₊L [defaults to 1.0]
  capacitor₊C [defaults to 1.0]
  source₊V [defaults to 1.0]
Observed (24): see observed(rlc_model)
```

Out[56]:

$$\frac{\mathrm{d}\texttt{capacitor.v}(t)}{\mathrm{d}t} = \frac{\texttt{capacitor.i}(t)}{\texttt{capacitor.C}} \tag{70}$$

$$\frac{\mathrm{d}\texttt{inductor.i}(t)}{\mathrm{d}t} = \frac{\texttt{inductor.v}(t)}{\texttt{inductor.L}} \tag{71}$$

In [57]:
```
observed(rlc_model)
```

Out[57]:

$$\texttt{ground.g.v}(t) = 0 \tag{72}$$
$$\texttt{source.v}(t) = \texttt{source.V} \tag{73}$$
$$\texttt{capacitor.n.v}(t) = -0 \tag{74}$$
$$\texttt{capacitor.p.v}(t) = \texttt{capacitor.v}(t) \tag{75}$$
$$\texttt{capacitor.i}(t) = \texttt{inductor.i}(t) \tag{76}$$
$$\texttt{source.n.v}(t) = \texttt{capacitor.n.v}(t) \tag{77}$$
$$\texttt{source.p.v}(t) = \texttt{source.v}(t) + \texttt{capacitor.n.v}(t) \tag{78}$$
$$\texttt{inductor.n.v}(t) = \texttt{capacitor.p.v}(t) \tag{79}$$
$$\texttt{inductor.p.i}(t) = \texttt{capacitor.i}(t) \tag{80}$$
$$\texttt{inductor.n.i}(t) = -\texttt{capacitor.i}(t) \tag{81}$$
$$\texttt{capacitor.p.i}(t) = \texttt{capacitor.i}(t) \tag{82}$$
$$\texttt{resistor.p.v}(t) = \texttt{source.p.v}(t) \tag{83}$$
$$\texttt{resistor.i}(t) = \texttt{inductor.p.i}(t) \tag{84}$$
$$\texttt{resistor.n.i}(t) = -\texttt{inductor.p.i}(t) \tag{85}$$
$$\texttt{capacitor.n.i}(t) = -\texttt{capacitor.p.i}(t) \tag{86}$$
$$\texttt{source.n.i}(t) = \texttt{resistor.i}(t) \tag{87}$$
$$\texttt{source.p.i}(t) = -\texttt{resistor.i}(t) \tag{88}$$
$$\texttt{ground.g.i}(t) = -\texttt{resistor.i}(t) + \texttt{capacitor.i}(t) \tag{89}$$
$$\texttt{resistor.v}(t) = \texttt{resistor.R}\texttt{resistor.i}(t) \tag{90}$$
$$\texttt{resistor.p.i}(t) = \texttt{resistor.i}(t) \tag{91}$$
$$\texttt{source.i}(t) = \texttt{source.p.i}(t) \tag{92}$$
$$\texttt{inductor.p.v}(t) = -\texttt{resistor.v}(t) + \texttt{source.p.v}(t) \tag{93}$$
$$\texttt{inductor.v}(t) = -\texttt{capacitor.v}(t) - \texttt{resistor.v}(t) + \texttt{source.v}(t) \tag{94}$$
$$\texttt{resistor.n.v}(t) = \texttt{inductor.p.v}(t) \tag{95}$$

In [58]:
```
u0 = [ rlc_model.capacitor.v => 0.0
       rlc_model.inductor.i => 0.0
]
```

Out[58]:
```
2-element Vector{Pair{Num, Float64}}:
 capacitor₊v(t) => 0.0
  inductor₊i(t) => 0.0
```

In [59]:
```
prob = ODEProblem(rlc_model, u0, (0, 10.0))
```

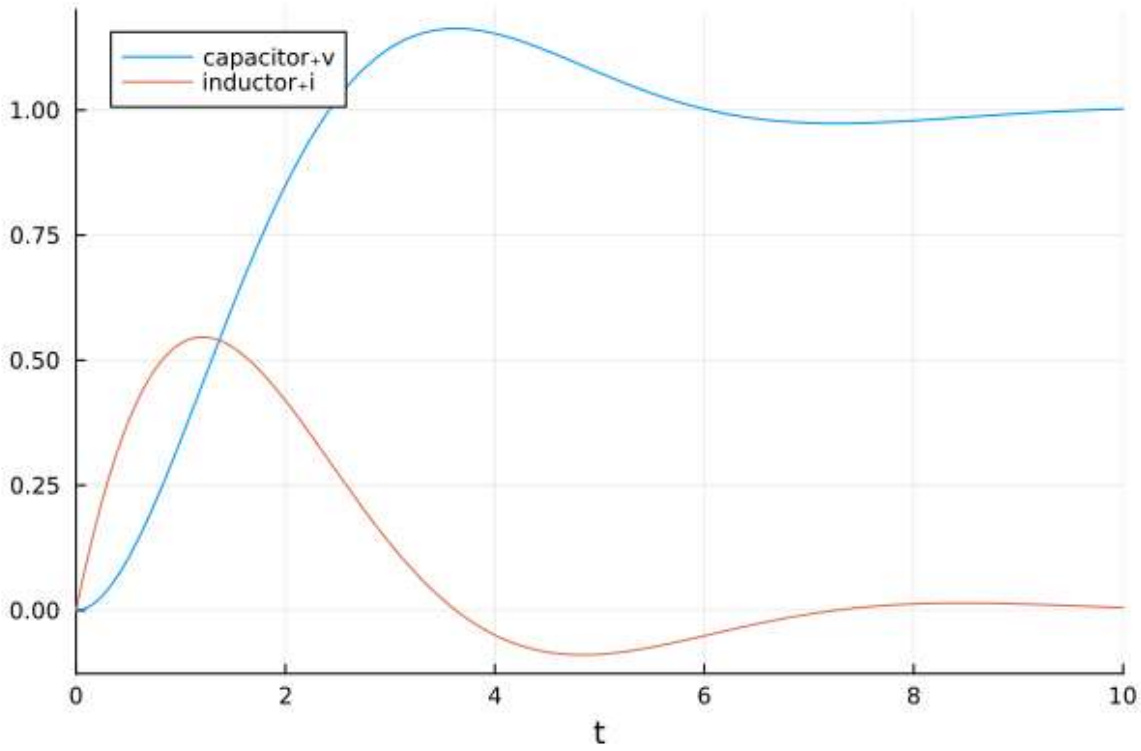
Out[59]:
```
ODEProblem with uType Vector{Float64} and tType Float64. In-place: true
Initialization status: FULLY_DETERMINED
Non-trivial mass matrix: false
timespan: (0.0, 10.0)
u0: 2-element Vector{Float64}:
 0.0
 0.0
```

In [62]:
```
sol = solve(prob)
```

Out[62]:
```
retcode: Success
Interpolation: 3rd order Hermite
t: 23-element Vector{Float64}:
  0.0
  9.999999999999999e-5
  0.0010999999999999998
  0.011099999999999997
  0.06011862644596693
  0.15042632519643986
  0.28380492936210605
  0.4678186118211677
  0.7152653512117267
  1.0347207106005918
  1.4313648280918627
  1.9003710804437082
  2.4488777227619627
  3.0778420261089807
  3.8238016789236675
  4.511767493047128
  5.20478873282933
  5.939653827614375
  6.737441224771508
  7.636538803204853
  8.455698732346535
  9.252425660114291
 10.0
u: 23-element Vector{Vector{Float64}}:
 [0.0, 0.0]
 [4.999833333333388e-9, 9.999500000000413e-5]
 [6.047781666800799e-7, 0.0010099395000060989 8]
 [6.13770629015738e-5, 0.011038395631125119]
 [0.0017709171509673331, 0.05831203956244671]
 [0.010747355599385363, 0.1391329785976398]
 [0.03647738129711264, 0.24378730937010124]
 [0.09253533051178127, 0.360201390056661]
 [0.1961891051410076, 0.4688260927491309]
 [0.35889500536174096, 0.5374947938657386]
 [0.57413092563836, 0.5338073355361069]
 [0.8063452434824447, 0.44523351164296776]
 [1.0089551505926229, 0.28934352374545014]
 [1.1339531960128406, 0.11358238282244829]
 [1.160098956787946, -0.028849553664129803]
 [1.1174663163732967, -0.08384692988675964]
 [1.0569695772902528, -0.0837688643397119]
 [1.0054577786449508, -0.05382453346213802]
 [0.9775947753910227, -0.01724513854127401]
 [0.9751095000256219, 0.008216345939254058]
 [0.9853487202688167, 0.014516694787007317]
 [0.9959610324860074, 0.011165794886770173]
 [1.0021660084365291, 0.005389025012217462]
```

In [63]:
```
plot(sol)
```

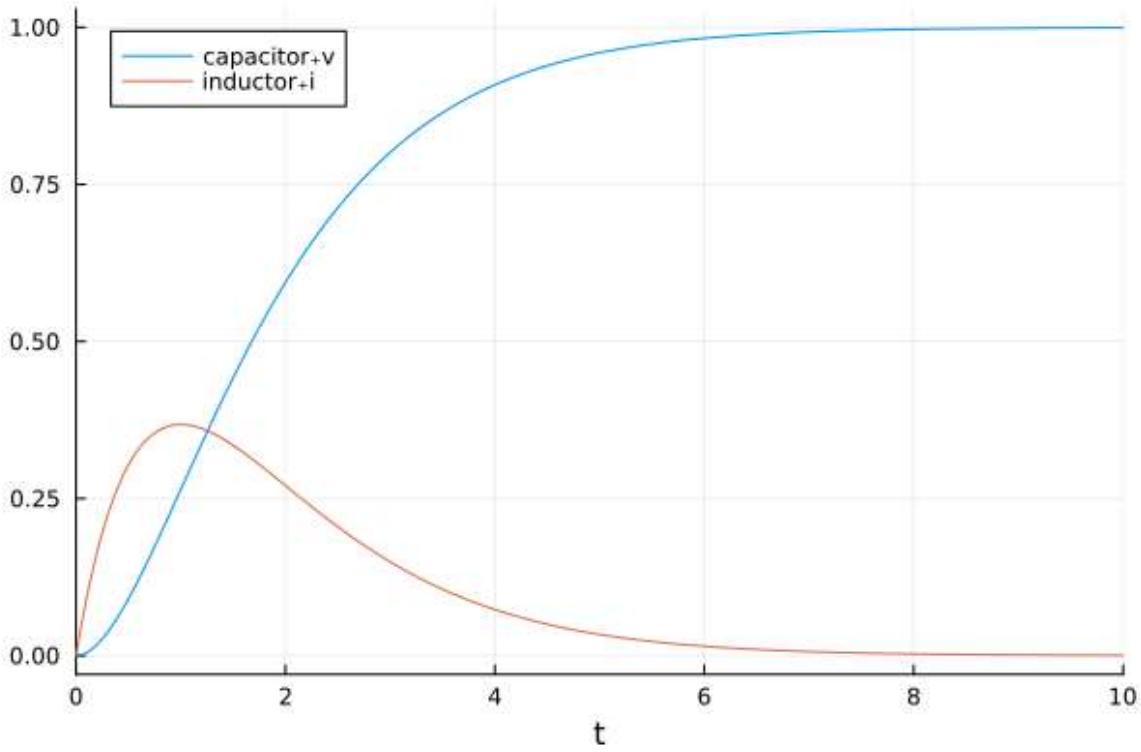
Out[63]:



Changing RLC model with resistance value

In [64]:
```
@mtkcompile rlc_model2 = RLCModel(resistor.R=2.0)
prob = ODEProblem(rlc_model2, u0, (0, 10.0))
sol = solve(prob)
plot(sol)
```

Out[64]:



Parallel RLC Circuit

In [72]:
```
@mtkmodel RLCParModel begin
    @description "A circuit with a constant voltage source in series with resistor,
    @components begin
        resistor = Resistor(R = 1.0)
        inductor = Inductor(L = 1.0)
        capacitor = Capacitor(C = 1.0)
        source = ConstantVoltage(V = 1.0)
        ground = Ground()
    end
    @equations begin
        connect(source.p, resistor.p)
        connect(resistor.n, inductor.p)
        connect(inductor.p, capacitor.p)
        connect(inductor.n, capacitor.n)
        connect(capacitor.n, source.n)
        connect(source.n, ground.g)
    end
end
;
```

In [70]:
```
@mtkcompile rlc_par = RLCParModel()
```

Out[70]:

$$\frac{\mathrm{d}\texttt{capacitor.v}(t)}{\mathrm{d}t} = \frac{\texttt{capacitor.i}(t)}{\texttt{capacitor.C}} \tag{96}$$

$$\frac{\mathrm{d}\texttt{inductor.i}(t)}{\mathrm{d}t} = \frac{\texttt{inductor.v}(t)}{\texttt{inductor.L}} \tag{97}$$

In [71]:
```
@mtkcompile rlc_par = RLCParModel(resistor.R=2.0)
u0 = [ rlc_model.capacitor.v => 0.0
       rlc_model.inductor.i => 0.0
]
prob = ODEProblem(rlc_par, u0, (0, 10.0))
sol = solve(prob)
plot(sol)
```

Out[71]: